

Edit Team 2016549 Project Proposal Evaluation

Record successfully updated

Executive Summary- contents *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Executive Summary - format *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Background and motivation *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Project Goal *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Project Requirements *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Validation and Acceptance Tests *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Possible Solutions and Design Alternatives	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input checked="" type="radio"/> Good	<input type="radio"/> Excellent
System Level Overview of Selected Design *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Module Level Descriptions of Selected Design *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Budget of Proposed Design *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input checked="" type="radio"/> Good	<input type="radio"/> Excellent
Assessment of Design Selected *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input checked="" type="radio"/> Good	<input type="radio"/> Excellent
Work Breakdown *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Gantt Chart *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Feasibility assessment *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input checked="" type="radio"/> Good	<input type="radio"/> Excellent
Summary of Goal Diagnosis	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent

and Solution Proposal *							
Writing	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input checked="" type="radio"/> Good	<input type="radio"/> Excellent
Content *	<input type="radio"/> Missing /Unable to evaluate	<input type="radio"/> Fails /Inadequate	<input type="radio"/> Marginal	<input type="radio"/> Adequate	<input type="radio"/> Satisfactory	<input type="radio"/> Good	<input checked="" type="radio"/> Excellent
Ethics review required?	<input checked="" type="radio"/> No	<input type="radio"/> Suggested	<input type="radio"/> Mandatory				
Safety review required?	<input checked="" type="radio"/> No	<input type="radio"/> Suggested	<input type="radio"/> Mandatory				
Evidence of development of document through iteration *	<input type="radio"/> Unable to evaluate	<input type="radio"/> Some	<input type="radio"/> As expected				
Comments to team	<p>Overall, this proposal is very strong. It is detailed and comprehensive. Although it shows good progress and grasp of the background, in sections such as the Technical Design, so much detail is presented, it becomes difficult to grasp the big picture and the general direction of the work. This wasn't the easiest proposal to grasp, but all the information is there. Keep up the good work.</p>						

Project Proposal ECE496 2016-2017

MovieQA: Evaluate Automatic Story Comprehension from both Video and Text

Team ID: 2016549

Supervisor Name: Raquel Urtasun

Administrator Name: Khoman Phang (Section 1)

Students:

Emily Vukovich

Soon Chee Loong

Yining Zhang

18th November 2016

Executive Summary



Movie Question-Answering or MovieQA is a supervised machine learning problem with the context of answering movie story related questions. This requires that the machine learns high level semantics regarding intent, motivation, and emotion by using movies as a condensed scenes of human life. It builds on previous machine learning tasks, particularly on VisualQA which combines natural language processing (NLP) and image captioning to answer questions about images. However, MovieQA is distinct from VisualQA in that it incorporates the temporal aspect of video, draws from both visual and textual sources to answer the question, and tests understanding of human reasoning rather than short descriptions. Available infrastructure for the MovieQA project includes the MovieQA dataset, question type breakdown, trained models used in the project, and graduate student researchers. The average benchmark is currently only around 35% accuracy. The goal of this project is to characterize text-based MovieQA models and build upon them to increase accuracy.

Given movie data (plot summaries, subtitles, script) and a set of multiple choice questions, the MovieQA design should output what it determines to be the most likely answer. The project design approach considers each of the various tasks required for MovieQA separately: word encoding. Natural Language Processing (NLP) architecture, and decision algorithms. Multiple architectures will be implemented for each task and their results will be compared and analysed.

The work schedule is broken into 7 phases: a general knowledge phase, background review, replication of MovieQA word encodings, Framework Building, Natural Language Processing, the Decision Module, and final overall adjustments. Each phase will involve analysis and/or presentation of results to ensure that technical requirements are met. The final design is expected to be completed by March 2017 and presented in April 2017. Presentation will include both architecture analysis results and demonstration of the final model. Estimated required funding for the project is \$0. Time is a major risk with this project as it is difficult to predict model training time; there is therefore a tradeoff between quality of a trained model and training time.

TABLE OF CONTENTS

1 PROJECT DESCRIPTION	1
1.1 Background & Motivation	1
1.2 Project goal	3
1.3 Project requirement	4
1.3.1 Functional Requirements	4
1.3.2 Constraints	4
1.3.3 Objectives	5
1.4 Validation tests	5
2 TECHNICAL DESIGN	6
2.1 Possible solutions & alternatives	6
2.1.1 Word Encoding	6
2.1.1.1 Term-Inverse Document Frequency (TF-IDF)	7
2.1.1.2 Word2vec	7
2.1.2 Natural Language Understanding	8
2.1.2.1 Recurrent Neural Network (RNN)	8
2.1.2.2 Skipthoughts	9
2.1.2.3 Long Short Term Memory (LSTM)	9
2.1.3 Decision Module	9
2.1.3.1 Neural Model Networks	9
2.1.3.2 Convolutional Neural Network (CNN)	10
2.2 System-Level Overview	11
2.3 Module Level Description	12
2.3.1 Word Representation Module	12
2.3.2 Database Module	12
2.3.3 Natural Language Understanding Module	12
2.3.4 Decision Module	13
2.3.5 Testing Module	13
2.4 Assessment of Proposed Solution	13
3 WORK PLAN	14
3.1 Work Breakdown Structure	14
3.2 Gantt Chart	21
3.3 Feasibility Assessment	22
3.3.1 Skills and Resources	22
3.3.1.1 Available Skills and Resources	22
3.3.1.2 Skills and Resources to be obtained	23
3.3.1.3 Risk Identification	23

3.3.1.4 Risk Mitigation	24
4 CONCLUSION	24
5 REFERENCES	24
6 APPENDICES	28
6.1 Appendix A: Student-Supervisor Agreement Form	28
6.2 Appendix B: Project Proposal Document Attribution Table	29
6.3 Appendix C: Glossary	31

1 PROJECT DESCRIPTION

1.1 Background & Motivation

Question-answering (QA) is a field of study dealing with teaching machines to answer questions posed in everyday language, or natural language [1]. It has many applications such as chat bots [2], dialogue systems [3], and information retrieval [4]. However in addition to answering text-based questions, a machine should, like humans, be able to answer questions based on an image [5]. According to [5], machine intelligence can be measured by Visual Question-Answering (VisualQA) [5].

VisualQA combines natural language and images to solve the QA problem: it uses generated captions to answer questions about images [6] [7]. However, most VisualQA algorithms have been shown to be myopic, jump to conclusions, and stubborn as they fail to generalize to novel instances as shown in Figure 1 [8]. A truly intelligent machine should infer high level semantics such as motivation, intent, and emotion [9].



Figure 1: VisualQA example, showing questions (Q), ground truth answers (GT A) and predicted answer (Predicted A). The algorithm answers 'cake' to the test sample because there is a cake in the image. This answer is unrelated to the semantic of the test sample being a wedding reception. [11]

Movie Question-Answering or MovieQA aims to improve machine understanding of human behaviour by training machines on semantic and emotional understanding of videos. MovieQA adds a time dimension to the VisualQA task, thereby testing the machine's ability to infer higher level understanding from a series of scenes. Ideally, it

would do this by combining information from text-based sources with the video sources to answer multiple-choice questions [9].

The MovieQA project in [9] investigated various approaches to and provided a basic framework for the MovieQA task. This involved the creation of the MovieQA dataset, which is comprised of approximately 15,000 question-answer text sets and various movie-related data. [12] Each question-answer set consists of a question and five labeled multiple choice answers (1 correct, 4 incorrect).


Movie	Harry Potter and the Chamber of Secrets	Amadeus	American Gangster
Question	What does Harry trick Lucius into doing?	Does Salieri admire Mozart's genius before he meets him in person?	How many years does Lucas serve in prison?
Story		<ul style="list-style-type: none"> - He was my idol. - Mozart. - I can't think of a time when I didn't know his name. 	<i>... provides evidence that leads to more than one-hundred further drug-related convictions, while he himself is sentenced to 70 years in prison, of which he serves 15 years and is released in 1991.</i>
Correct answer	Freeing Dobby	Yes, he thinks his talent is a God's gift	15 years
Wrong answer 1	Releasing Dobby to Harry's care	He doesn't think that Mozart is really a genius	70 years
Wrong answer 2	Releasing Dobby to Dumbledore's care	He thinks Mozart is totally overrated	50 years
Wrong answer 3	Releasing Dobby to Hagrid's care	No, he thinks Mozart is good but not a gift from God	17 years
Wrong answer 4	Admitting he put Tom Riddle's diary in Ginny's cauldron	He does not know anything about Mozart so he does not know whether he is good or not	35 years

Figure 2: Example of Question-Answer sets in the MovieQA dataset as well as corresponding data snippets. (The story displayed only shows a part of the given plot/video/script relevant to the question, the model will have access to the entire data and must infer the relevant parts on its own.) [10]

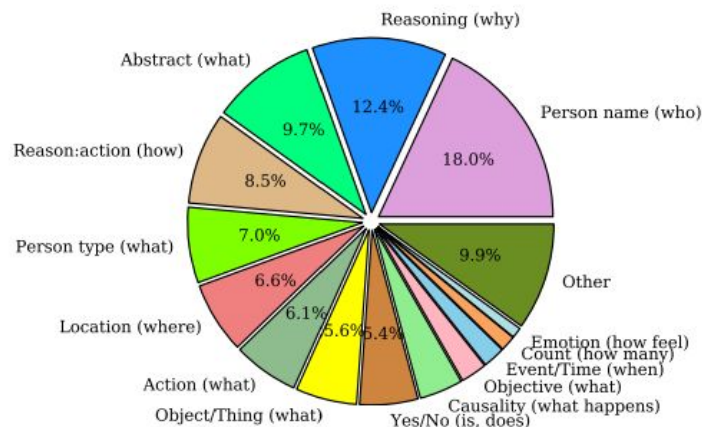
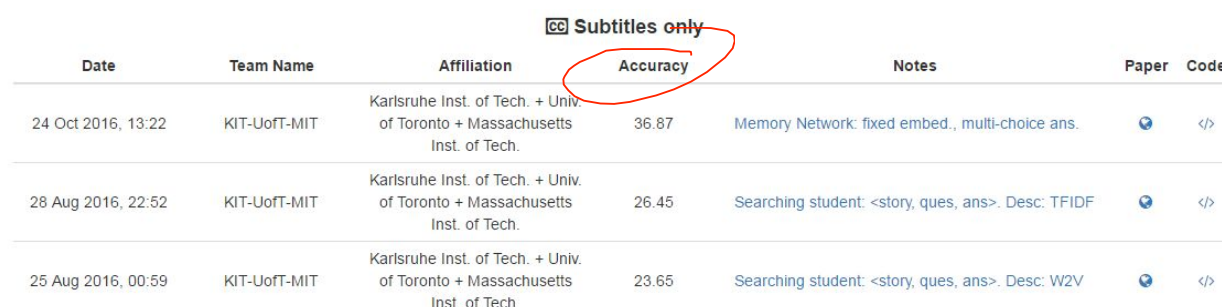


Figure 3: Breakdown of question types in MovieQA dataset. Different questions test different types of understanding. For example, “What happens” questions test causality and “What did X do?” questions test actions. [12]

The movie-related data is either text-based (plot summaries, subtitles, movie scripts, descriptive video-service) or video-based (movie clips that are several minutes in duration). (See glossary, Appendix C for descriptions of data) This project also involved the creation of the online MovieQA benchmark, which runs trained models on additional MovieQA test data inaccessible to the public, and provides an accuracy score. Submitted attempts are ranked by accuracy.



Date	Team Name	Affiliation	Accuracy	Notes	Paper	Code
24 Oct 2016, 13:22	KIT-Uoft-MIT	Karlsruhe Inst. of Tech. + Univ. of Toronto + Massachusetts Inst. of Tech.	36.87	Memory Network: fixed embed., multi-choice ans.		
28 Aug 2016, 22:52	KIT-Uoft-MIT	Karlsruhe Inst. of Tech. + Univ. of Toronto + Massachusetts Inst. of Tech.	26.45	Searching student: <story, ques, ans>. Desc: TFIDF		
25 Aug 2016, 00:59	KIT-Uoft-MIT	Karlsruhe Inst. of Tech. + Univ. of Toronto + Massachusetts Inst. of Tech.	23.65	Searching student: <story, ques, ans>. Desc: W2V		

Figure 4: Example of MovieQA benchmark leaderboard. Here, submission results using only movie subtitles are shown. [11]

Current implementations do not use the audio data from the movie clips and rely on combinations of image data from clips and text-based data [9]. The MovieQA project implemented a number of text and video representations: TF-IDF, word2vec, skip-thoughts, and a story-sentence mapping to a scene in a video [12]. Average accuracy is around 35% [9].

1.2 Project goal

The project goal is to develop, and characterize the performance of different word encoding and natural language processing algorithms in the context of the MovieQA task. Computer vision algorithms may also be considered, however these will not be the focus of the project.

1.3 Project requirements

1.3.1 Functional Requirements

- A. The design should be modifiable during training. (i.e. it should be possible to temporarily stop training, modify its internal architecture, and resume training)
- B. The design should take as inputs from the question-answer text set, the text-based story data (e.g., plot) (and possibly video) and produce a selected answer from the multiple choice question-answer set as output [9] :
 - a. The design should produce a set of word encodings given the question-answer text set and the text-based story data
 - b. The design should produce a representation of sequences of words (e.g., paragraphs) which captures semantic significance of the sentences (see glossary, Appendix C)
 - c. The design should select an answer from the question-answer text set based on the sequence representation of the question-answer set and the text-based story
- C. The design should be trained to convergence (the accuracy fluctuates within 5%).
- D. The design should demonstrate natural language comprehension abilities such that it can comprehend the MovieQA question-answer text sets and the text-based story data.
- E. The design should be trained on the MovieQA dataset in addition to larger datasets such as text8, a dataset which contains large amounts of text pulled from Wikipedia [13]; the design vocabulary should therefore be limited to the words found in these datasets..

1.3.2 Constraints

- A. The algorithm must not access the test set during training.

You do not define the three different data sets (training set, test set, and validation set) and their roles in machine learning

1.3.3 Objectives

- A. Maximize answer accuracy: At test time, higher accuracy is better. (Note: the current benchmark is approximately 47% accuracy, using plot synopsis only [11])
- B. Minimize training time: When training, convergence should occur in a relatively short time (less than a week).
- C. Minimize inference time: Reduce time taken for test set prediction.

✓ **1.4 Validation tests** The basic system validation test described in draft B was good and can be kept because it represented testing at the highest level. The tests below represent the next level of detail and are complementary.

Validation Test	Level	Test Procedure	Success Criteria
Loss Plotting	Module & System	Plot training and <u>validation loss</u> against number of back-propagation steps during training on the required datasets.	Must converge within accuracy bounds of 5%.
QA Analysis	System	Show the QA results of the model when run on the <u>validation set</u> and analyze predictions (based on question type as shown in Figure 3, question and answer length, similarity between question and answer) to show characteristics of language comprehension.	Must show characteristics and intelligence of model through data. Prepare to interpret the results for the reader.
Accuracy Validation	System	Select the set of hyperparameters that produces the highest validation accuracy results for promising models, and submit those to the online MovieQA benchmark for test accuracy rating[9].	Confirm that validation accuracy deviates from test accuracy by less than 7%.
Training Time Test	Module	Record wallclock timestamps when training to determine time taken.	Less than 168 hours (1 week) to train model.

Inference Time Test	System	Time inference tasks for final models on validation dataset with the same machine to measure seconds taken per question.	Less than 30sec per question (less than 5.2 days to infer entire dataset of 15000 Qs).
---------------------	--------	--	--

Presentation of final results will depend on the number of answering sources tested (subtitles, Descriptive Video Service, etc.; see glossary in Appendix C). A display might include a video clip with the accompanying text source, question, and answer, followed by the machine's answer selection. ✓

2 TECHNICAL DESIGN This section is almost too detailed. Although it shows good progress and grasp of the background, it makes it difficult to grasp the big picture and obscures the general direction of the team.

2.1 Possible solutions & alternatives

Potential solutions for this project can be broken down into its functions: word encoding, building representations for collections of words (i.e. sentences) and decision making for answer selection (Section 2.2 shows how these fit together). Some models can cover multiple functions. Multiple models will be implemented in each area to obtain experimental evidence to compare performance based on the validation tests. Note that more detailed descriptions can be found in Appendix C.

2.1.1 Word Encoding

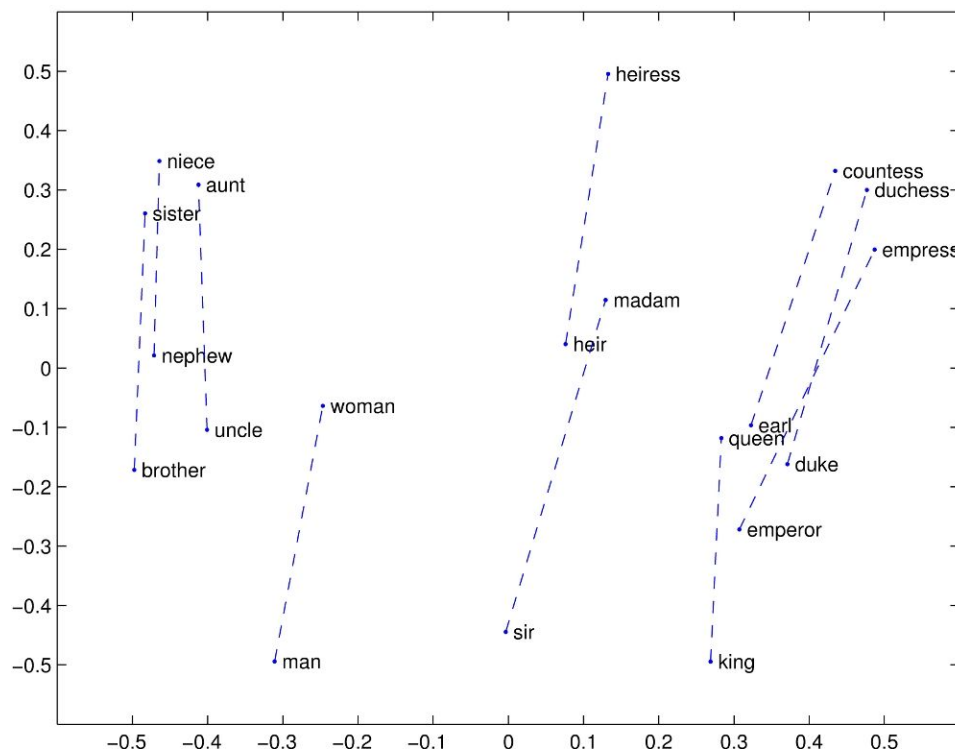
To comprehend natural language questions and answers, the machine must have word representations with an idea of semantic similarity. This is achieved by finding numerical representations of words as vectors, where the method in which the vectors are constructed determines the kind of semantics captured in the encoding (e.g., an encoding might differentiate between male and female based on the distance between vectors). The vector representation of words presents text in a format that allows computations to be performed on sequences of words (such as sentences and paragraphs), thereby facilitating the Natural Language Understanding models.

2.1.1.1 Term-Inverse Document Frequency (TF-IDF)

TF-IDF is commonly used in search ranking and document subject labeling [14]. It finds an overall score for a word based on how often the word appears in a given document (TF), and how many documents the word appears in (IDF).

2.1.1.2 Word2vec

Word2vec uses a neural network to map words to vectors such that similar words are closer together. It produces sets of weights, which are the vector representations of the words. Word2vec is particularly successful at capturing semantic similarity as shown in Figure 5 [15].



✓
nice
illustration
of concept

Figure 5: Visual representation of semantic similarity. Words are positioned according to their word vector representations. Words with similar relationships have similar distances between them, such as the man-woman, brother-sister, king-queen pairs [16]

The most common word2vec implementations are CBOW and skip-gram: Skip-gram predicts a selected word using those surrounding it while CBOW predicts surrounding words using their single center word as shown in Figure 6 [17]–[19].

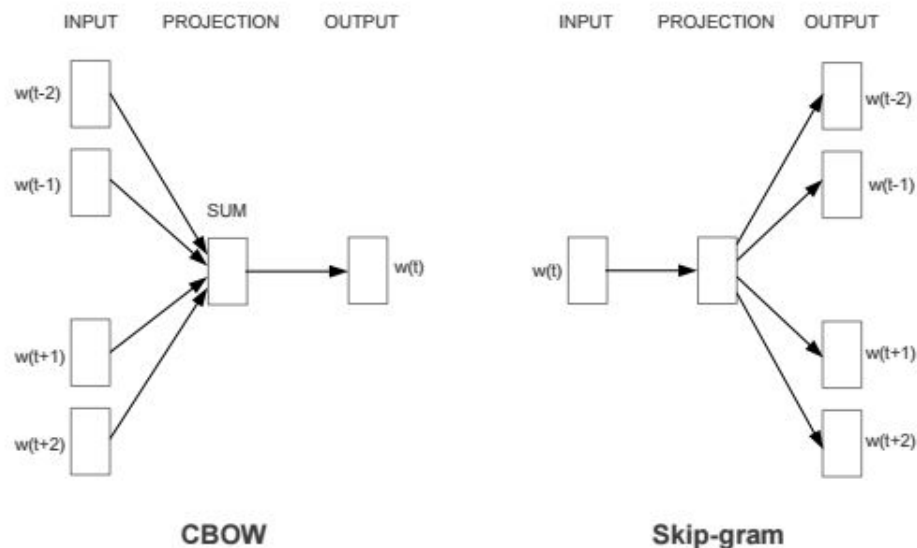


Figure 6: CBOW and Skip-gram word2vec models showing prediction using/for 2 previous and 2 following words, where $w(t)$ is the current word, $w(t-1)$ is the previous word, and $w(t+1)$ is the following word. CBOW uses surrounding words to predict the current word; Skip-gram predicts surrounding words using the current word [15]

2.1.2 Natural Language Understanding

These models produce vectors that represent an understanding of natural language. This allows the design to interpret long word sequences and perform computations such as similarity between two sentences [20]. This enables the decision model to understand the question, answers, and movie text to decide the answer.

2.1.2.1 Recurrent Neural Network (RNN)

A plain recurrent neural network uses a single hidden layer with a non-linear activation function, such as a sigmoid, to determine the output. These are useful when dealing with sequence inputs and outputs; however due to the effect of vanishing

gradient as described in [21], only inputs within several timesteps will affect the current output.

2.1.2.2 Skipthoughts

Skipthoughts extends word2vec skip-gram models to entire sentences [22]. Here an RNN is used to build on the word2vec RNN, resulting in an encoding of a sentence.

2.1.2.3 Long Short Term Memory (LSTM)

LSTM is a type of RNN which uses a cell state to enable longer memory dependence [23]. It implements an additive state change and so mitigates the effect of vanishing gradient. It is therefore useful for longer sequences.

2.1.3 Decision Module

The decision module is where the scoring function to select an answer is determined. It is possible to implement a naive decision function without using a neural network. For instance, cosine similarity might compute the dot product of sentence vectors to find the best match between the story data and the question. However, using a Neural Network (NN) to determine the decision function will ideally produce better results than such naive implementations.

2.1.3.1 Neural Model Networks

NMNs assemble different neural architectures depending on the question semantics [24]. In VisualQA, this tends to perform worse than MemN2N, however, in movies, this could speed up training time by reducing network complexity in certain cases. The architecture could be expanded to take the long form multiple choice answers into account.

2.1.3.2 End to End Memory Networks (MemN2N)

MemN2N was originally built for language processing [25], with the core idea of giving the network large datasets at its disposal (i.e. dictionaries). The MovieQA paper

used it generally in that context with only minor modifications for video input [25]. In fact, giving the MemN2N network video data actually decreased accuracy [25].

2.1.3.2 Convolutional Neural Network (CNN)

Convolutional neural networks use series of filters convolved with the original input to produce convolutional layers. The filters determine the feature of interest to be observed; each of the convolutional layers therefore holds information about the corresponding filter feature.

2.1.3.4 RNN-CNN paired architecture

These use both RNN and CNN for training. However, the CNN should be (pre)trained to detect objects, specific entities and actions. The reasoning for this architecture is that there is insufficient data to train the net in its entirety in the case of video input; however given that video is not the focus of this project, this is not a major concern.

2.2 System-Level Overview

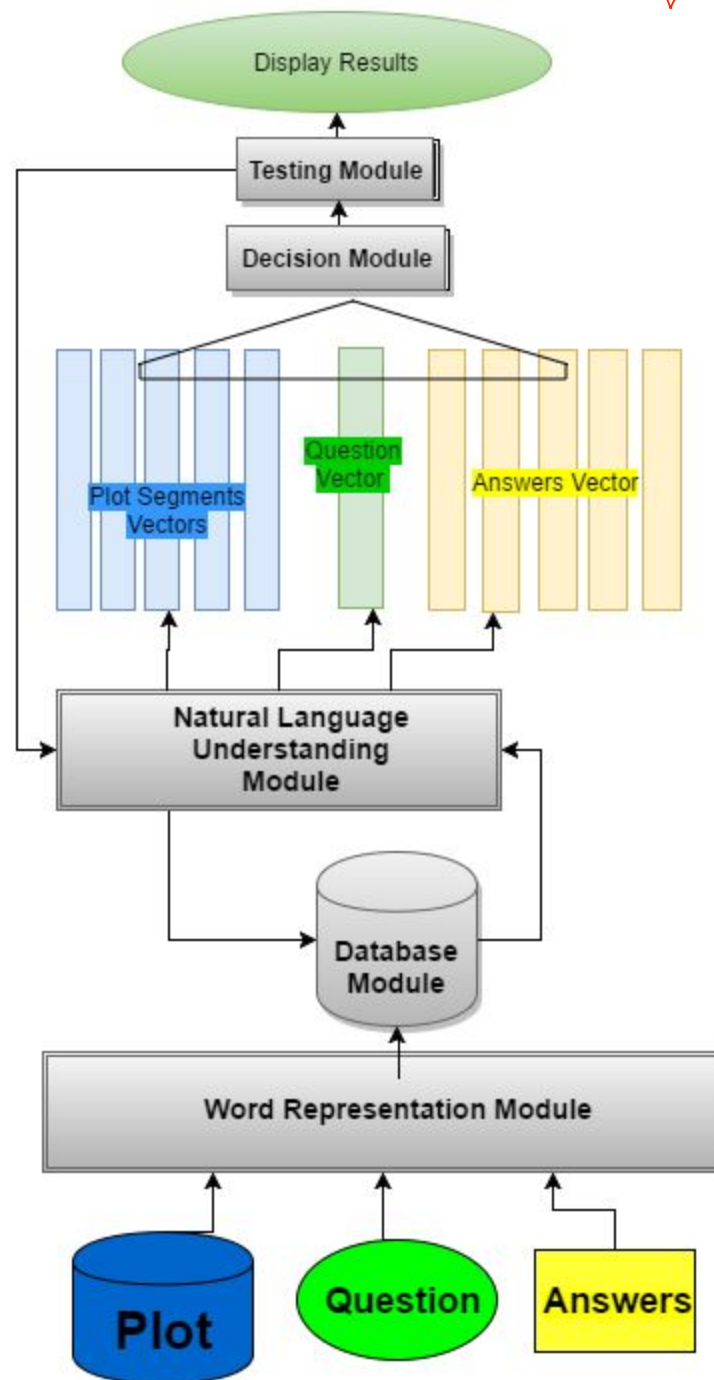


Figure 7: System Level Overview.

2.3 Module Level Description



These describes in the detail the modules found in Figure 7.

2.3.1 Word Representation Module

Input: Blocks of raw ascii text (i.e. plot, questions, answers, books, wikipedia articles)
Output: <ul style="list-style-type: none">- Mapping from a finite set of distinct words in vocabulary (vocab) to vector embeddings.- Tokenized representation of input text.
Function: <p>Cleans input data (e.g. word stemming) and runs algorithm to construct vectors for a selected finite set of words called vocabulary. The dimension of the embedding and the size/contents of the vocab will be selected in this module. In other words, it maps each individual word to vectors. We will experiment with a combination of TF-IDF and word2vec and possibly other models like GloVe[25].</p>

2.3.2 Database Module

Input: <ul style="list-style-type: none">- Mapping from vocabulary word to vector embedding.- File location of database.
Output: Serialize vector representation of words and text.
Function: Periodically stores word vectors that are in-training or fully trained into disk to be able to reuse them for different word collection and decision algorithms. Serves as a fault tolerance during training. Allows design to be modifiable during training.

2.3.3 Natural Language Understanding Module

Input: <ul style="list-style-type: none">- Mapping from a set of vocabulary words to vector embedding.- Tokenized representations of plot, questions and answer text.

Output: Vector representation of segments of text (ie. plot sentences, question).
--

Function: Devise vector representations of sets of words. It maps sentences to vectors. Potentially learns the optimal word set length to represent ideas. Simple sum of words representations will be benchmarked against more complicated skip-thought/RNN/LSTM representations.

2.3.4 Decision Module

Input: Vector representation of text (data, question and answers) segments.
--

Output: Answer choice.

Function: Select a final answer choice given all the data in vector format. This module will be trained on the MovieQA training data. Different cost functions and neural network architectures will be tested for the final decision module and compare the results with a standard cosine similarity function.

2.3.5 Testing Module

Input: Predicted Answers

Output: Display the list of wrongly answered questions.
--

Function: This module displays the set of correctly chosen answers as well as the set of wrongly chosen answers with their respective questions and possible answer choices. It also sends results back to the Natural Language Understanding Module for supervised learning.
--

2.4 Assessment of Proposed Solution

The proposed solution is to build each word encoding, devise a few architectures that incorporate the natural language processing and possibly computer vision components described, and combine and benchmark them against our data. In order to characterize the different algorithms and their effectiveness for the MovieQA task, a number of different architectures will be implemented and compared experimentally.

Research will be conducted into the performance of each architecture for various tasks prior to implementation. This research will be summarized in a standard framework the team will create such that the benefits, drawbacks, and unique characteristics of the architectures for specific tasks can be easily compared. In addition, following implementation, the performance of each architecture will be analyzed as outlined in section 1.4, both to assess the architectures' individual performance in MovieQA as well as to help determine how to implement the submodules outlined in section 2 (i.e., which word encoding modules to fit with which natural language processing module to fit with which decision module).

The modular structure of our proposed solution allows us to modify and train each component separately. This allows us to build up experience with machine learning without spending large amounts of time waiting for the network to train. It also gives the flexibility to train the entire network together if desired.

3 WORK PLAN

The process is broken down into four phases. One background review phase followed by three design phases. The background review phase requires replication of existing work as it is a learning process. Each design phase is similar in structure and components are detailed in section 3.1.

3.1 Work Breakdown Structure

Members will each be responsible for the specifics and implementation of one component (detailed below) for each design phase. The team will combine the components in each training/evaluation time block.

R = Responsible, A = Assist

Mathematical Proof = Derive equations for the algorithm.

Performance Evaluation = Evaluate performance of the algorithm based on our project requirement by reading research papers on existing applications. We will come up with a performance table to be filled in the future.

Implementation = Implement algorithm as a stand-alone tool which works on any existing applications.

Performance Analysis = Analyze the performance of the algorithm for our system and figure out a few plausible ideas to improve the performance. Provide 1 page report including results from validation tests.

Parameter Tweaking = Update parameters to improve accuracy for final presentation.

Very thorough breakdown

Task #	Task	Yining Zhang Annie	Soon Chee Loong	Emily Vukovich
	<u>General Knowledge</u>			
	→ Neural Networks ◆ Implement.	R		
	→ Universality Theorem ◆ Explain why neural network can learn any function in theory. ◆ Mathematical Proof: Activation Function		R	
	→ Cross Entropy Loss ◆ Explain how cross entropy loss is better than L1 and L2 loss for Neural Networks. ◆ Implementation.			R
	<u>Background Review</u>			
	→ Neural Network ◆ Mathematical Proof: Back-Propagation Equations	R		

	<ul style="list-style-type: none"> ◆ Provide a 10 minute lecture presentation to supervisor. 			
	→ Convolutional Neural Network <ul style="list-style-type: none"> ◆ Provide a 10 minute lecture presentation to supervisor. 		R	
	→ Recurrent Neural Network <ul style="list-style-type: none"> ◆ Provide a 10 minute lecture presentation to supervisor. 			R
	<u>Replicate Original Word Encodings</u>			
1	→ Word2Vec <ul style="list-style-type: none"> ◆ Implement a single layer word2vec. ◆ Update single layer word2vec to multiple layers ◆ Implement hierarchical softmax ◆ Implement negative sampling ◆ Apply code to MovieQA Dataset ◆ Match actual benchmark accuracy of 45% on plots. ◆ Provide 1 page report including results from validation tests 	R		
2	→ TF-IDF <ul style="list-style-type: none"> ◆ Implement Term Frequency Code. ◆ Implement Inverse 		R	

	<p>Document Frequency Code</p> <ul style="list-style-type: none"> ◆ Apply TF-IDF code to MovieQA dataset. ◆ Match actual benchmark accuracy of 47% on plots. ◆ Performance Analysis. 			
3	<p>→ Skip-thoughts</p> <ul style="list-style-type: none"> ◆ Implement non-optimized word2vec ◆ Build training batches, cross-validation sets ◆ Combine the RNN with word2vec ◆ Apply code to MovieQA dataset ◆ Match actual benchmark accuracy of 31% on plots (optimization) ◆ Performance Analysis. 			R
	<u>Build Frameworks</u>			
1	<p>→ Display Questions</p> <ul style="list-style-type: none"> ◆ Display correctly answered questions ◆ Display wrongly answer questions ◆ Update display to display chosen answer in green if correct, red if wrong. 	R		
2	<p>→ Clean Dataset</p> <ul style="list-style-type: none"> ◆ Remove non-alphanumeric characters 		R	

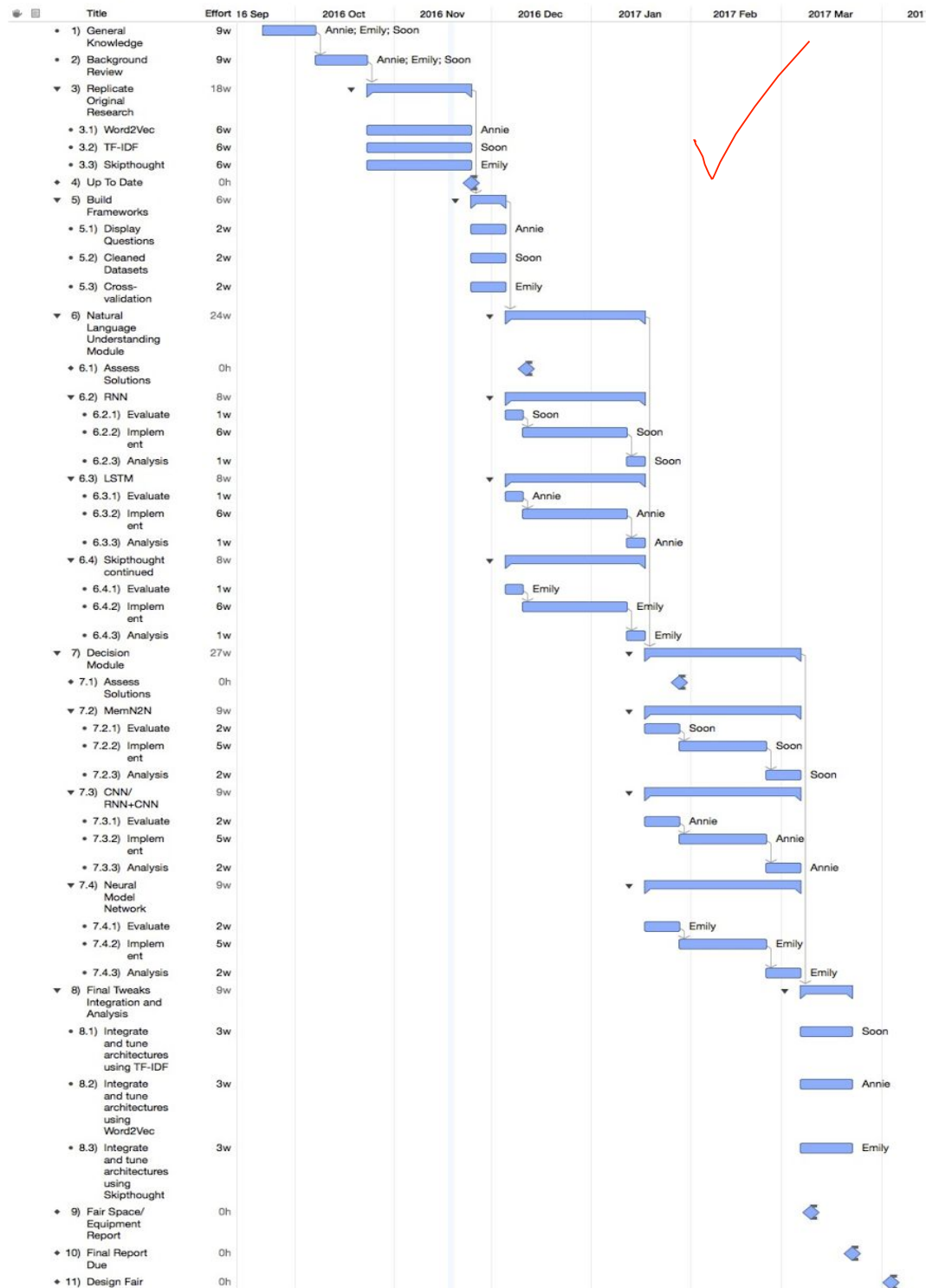
	<ul style="list-style-type: none"> ◆ Implement lowercase. ◆ Implement stemming of words. 			
3	<p>→ Cross Validation</p> <ul style="list-style-type: none"> ◆ Separate data into training set, validation set, and test set. ◆ Run algorithm across separated data. ◆ Iteratively permute the way the data are separated and run algorithms 			R
4	<p>→ Set up Database</p> <ul style="list-style-type: none"> ◆ Create python module to save data to disk ◆ Create python module to retrieve data from disk 		R	
	<u>Natural Language Understanding</u>			
1	<p>→ Recurrent Neural Network</p> <ul style="list-style-type: none"> ◆ Performance Evaluation ◆ Implementation. ◆ Integrate into the system. ◆ Detailed Analysis. ◆ Performance Analysis 		R	
2	<p>→ Long Short Term Memory</p> <ul style="list-style-type: none"> ◆ Performance Evaluation. ◆ Implementation. ◆ Integrate into the system. ◆ Performance Analysis. 	R		
3	<p>→ Skip-thoughts</p> <ul style="list-style-type: none"> ◆ Performance Evaluation. 			R

	<ul style="list-style-type: none"> ◆ Implementation. ◆ Integrate into the system. ◆ Performance Analysis. 			
	<u>Decision Module</u>			
1	→ End to End Memory Networks <ul style="list-style-type: none"> ◆ Performance Evaluation. ◆ Implementation. ◆ Performance Analysis. 	A	R	A
2	→ Convolutional Neural network + Recurrent Neural Network <ul style="list-style-type: none"> ◆ Performance Evaluation. ◆ Implementation for CNN. ◆ Implementation for RNN. ◆ Integrate CNN and RNN ◆ Integrate into the system. ◆ Performance Analysis. 	R	A	A
3	→ Neural Model Network <ul style="list-style-type: none"> ◆ Performance Evaluation. ◆ Implementation. ◆ Integrate into the system. ◆ Performance Analysis. 	A	A	R
	<u>Final Tweaks, Integration, Analysis</u>			
1	→ Word2Vec <ul style="list-style-type: none"> ◆ Integrate into the system. ◆ Parameter Tweaking. ◆ Performance Analysis. 	R		
2	→ TF-IDF <ul style="list-style-type: none"> ◆ Integrate into the system. ◆ Parameter Tweaking. ◆ Performance Analysis. 		R	

3	<p>→ Skip-thoughts</p> <ul style="list-style-type: none"> ◆ Integrate into the system. ◆ Parameter Tweaking. ◆ Performance Analysis. 			R
---	---	--	--	---

3.2 Gantt Chart

MovieQA: Gantt Chart



3.3 Budget Table

Item	Priority	Cost/unit	Quantity/#Hours	Total Cost	Requires Funding
Capital Equipment					
Computers	1	\$1,000		\$3,000	n
GPU Processing Power	3	\$0.13 /kWh	165W*6 months*15hrs/day = 445.5kWh	\$58	n
Internet Access	1	\$40 per Month	\$40*8months = \$320	\$320	n
Total Capital Equipment				\$3,378	
Software					
Datasets	1	\$0		\$0	n
Libraries (open-source)	1	\$0	unknown	\$0	n
Pre-trained models	2	\$0	unknown	\$0	n
Total Software				\$0	
Student Labour					
	Cost/unit	# Expected Hours	Total Cost		
Student 1	\$8	600	\$4,800		
Student 2	\$8	600	\$4,800		
Student 3	\$8	600	\$4,800		
Graduate Student assistance	\$14	56	\$15,184		
Total Student Labour (unfunded)			\$29,584		
Total Cost of Project				\$32,962	
Total Cost Requiring Funding				\$0	
Funding					
Students (\$0 ea)	\$0				
Supervisor	\$0				
Other (specify)	\$0				
Request from Design Centre	\$0				
Total Funding	\$0				

3.3 Feasibility Assessment

3.3.1 Skills and Resources

3.3.1.1 Available Skills and Resources

Libraries and pre-trained models - Open-source libraries like Numpy and TensorFlow [26] and pre-trained networks like the inception V3 network [27] will be needed. Most of these are publicly available. These tools are the current industry standard.

Datasets: Training and testing will require huge amounts of cleaned and labeled data. This will include the MovieQA [1] dataset which is freely available. Other, larger datasets such as text8 may also be used to provide more training data for the word2vec and Natural Language Processing modules [28].

Graduate students: Two of the students (Makarand Tapaswi and Yukun Zhu) who developed the original MovieQA project are available to assist with the design.

3.3.1.2 Skills and Resources to be obtained

Machine learning knowledge - Background knowledge on current state of the art as well as ability to code and train models. It will take several weeks to obtain sufficient knowledge.

CUDA Knowledge - The ability to program for GPUs may or may not be needed depending on the libraries we use [18].

Computer Power - Computers with high memory GPUs to train models. If university resources are used, it must be booked in advance as other students will also be using the lab space.

3.4.2 Risk

3.3.1.3 Risk Identification

- Insufficient training time: Significant, unexpected increases in training time can delay the rest of the project as the model can only be evaluated after training.
- Disruptions during training (e.g. power outage): This is especially important as disruptions might require training the model from scratch, which could take days or even weeks to complete [29]
- Overfitting: A model which fits the training data exactly is not necessarily ideal for prediction (i.e., it might fit points exactly but also fits to outliers and does not indicate the overall trend).
- This is a high risk, high reward project. Many architectures are difficult to implement, however, reaching a new benchmark can result in a published paper according to our supervisor.

3.3.1.4 Risk Mitigation

- To manage insufficient training time, we can reduce training data size. However, this may run into the risk of insufficient data for training or overfitting.
- ✓ - To manage possible disruptions during training, the state of the model should be periodically stored to a disk.
- To reduce overfitting, we can monitor validation accuracy curves and increase the weight of regularization algorithms like L2 regularization or Dropout [30]
- ✓ - If the project does not result in a higher accuracy benchmark, the analysis of why the architectures used were unsuccessful is still valuable research information.

4 CONCLUSION

The project aims to improve the overall accuracy of the MovieQA benchmark by building on previously used architectures. The design is broken into various sub-components: word encodings, language processing, and decision algorithms. The design process will involve implementing several models for each sub-component and comparing results when combining them into the overall MovieQA architecture. Models which are generally more successful for its given task will be given priority in implementation. Success of the project will be based on the design's performance relative to current MovieQA benchmarks.

5 REFERENCES **Information still incomplete (from draft B)**

- [1] “http://cs224d.stanford.edu/lecture_notes/notes1.pdf,” 20-Sep-2016. [Online]. Available: http://cs224d.stanford.edu/lecture_notes/notes1.pdf. [Accessed: 20-Sep-2016].
- [2] J. Jia and W. Chen, “Motivate the Learners to Practice English through Playing with Chatbot CSIEC,” in *Technologies for E-Learning and Digital Entertainment*, Springer Berlin Heidelberg, 2006, pp. 180–191.
- [3] R. Inoue, Y. Kurosawa, K. Mera, and T. Takezawa, “A question-and-answer

classification technique for constructing and managing spoken dialog system,” in *2011 International Conference on Speech Database and Assessments (Oriental COCOSDA)*, 2011, pp. 97–101.

- [4] L. HIRSCHMAN and R. GAIZAUSKAS, “Natural language question answering: the view from here,” *Nat. Lang. Eng.*, vol. 7, Feb. 2002.
- [5] D. Batra, A. Agrawal, S. Antol, M. Mitchell, C. L. Zitnick, and D. Parikh, “Measuring Machine Intelligence Through Visual Question Answering,” Aug. 2016.
- [6] M. Ren, R. Kiros, and R. Zemel, “Exploring Models and Data for Image Question Answering,” May 2015.
- [7] A. Agrawal *et al.*, “VQA: Visual Question Answering,” May 2015.
- [8] A. Agrawal, D. Batra, and D. Parikh, “Analyzing the Behavior of Visual Question Answering Models,” Jun. 2016.
- [9] M. Tapaswi, Y. Zhu, R. Stiefelbogen, A. Torralba, R. Urtasun, and S. Fidler, “MovieQA: Understanding Stories in Movies through Question-Answering,” Dec. 2015.
- [10] “MovieQA - Examples,” 18-Nov-2016. [Online]. Available: <http://movieqa.cs.toronto.edu/examples/>. [Accessed: 18-Nov-2016].
- [11] “MovieQA - Leader Board,” 20-Sep-2016. [Online]. Available: <http://movieqa.cs.toronto.edu/leaderboard/>. [Accessed: 20-Sep-2016].
- [12] Y. Zhu *et al.*, “Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 19–27.
- [13] “About the Test Data,” 18-Nov-2016. [Online]. Available: <http://mattmahoney.net/dc/textdata>. [Accessed: 18-Nov-2016].
- [14] “Tf-idf :: A Single-Page Tutorial - Information Retrieval and Text Mining,” 06-Oct-2016. [Online]. Available: <http://www.tfidf.com/>. [Accessed: 06-Oct-2016].
- [15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *arXiv ePrints*, pp. 1–12, Jan. 2013.
- [16] “GloVe: Global Vectors for Word Representation,” 18-Nov-2016. [Online]. Available:

Missing the
names of
the sources]

- <http://nlp.stanford.edu/projects/glove/>. [Accessed: 18-Nov-2016].
- [17] “Word2Vec Tutorial - The Skip-Gram Model · Chris McCormick,” 06-Oct-2016. [Online]. Available: <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>. [Accessed: 06-Oct-2016].
- [18] “http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_I_The_Skip-Gram_Model.pdf,” 06-Oct-2016. [Online]. Available: http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_I_The_Skip-Gram_Model.pdf. [Accessed: 06-Oct-2016].
- [19] “http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_II_The_Continuous_Bag-of-Words_Model.pdf,” 06-Oct-2016. [Online]. Available: http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_II_The_Continuous_Bag-of-Words_Model.pdf. [Accessed: 06-Oct-2016].
- [20] O. Levy and Y. Goldberg, “Linguistic Regularities in Sparse and Explicit Word Representations,” in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 2012, pp. 171–180.
- [21] M. A. Nielsen, “Neural Networks and Deep Learning,” 2013.
- [22] R. Kiros *et al.*, “Skip-Thought Vectors,” pp. 11–11, Jun. 2015.
- [23] “Recurrent Neural Network Tutorial, Part 4 – Implementing a GRU/LSTM RNN with Python and Theano – WildML,” 06-Oct-2016. [Online]. Available: <http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/>. [Accessed: 06-Oct-2016].
- [24] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel, “Visual Question Answering: A Survey of Methods and Datasets,” Jul. 2016.
- [25] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, “End-To-End Memory Networks,” *Advances in Neural Information Processing Systems*. pp. 2431–2439, 31-Mar-2015.

- [26] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” Mar. 2016.
- [27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” Dec. 2015.
- [28] “About the Test Data,” 18-Nov-2016. [Online]. Available: <http://mattmahoney.net/dc/textdata>. [Accessed: 18-Nov-2016].
- [29] A. Kulakov, M. Zwolinski, and J. Reeve, “Fault Tolerance in Distributed Neural Computing,” Sep. 2015.
- [30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2012.
- [31] “http://cs231n.stanford.edu/slides/winter1516_lecture10.pdf,” 06-Oct-2016. [Online]. Available: http://cs231n.stanford.edu/slides/winter1516_lecture10.pdf. [Accessed: 06-Oct-2016].
- [32] “Understanding LSTM Networks -- colah’s blog,” 06-Oct-2016. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed: 06-Oct-2016].

6 APPENDICES

6.1 Appendix A: Student-Supervisor Agreement Form



ECE496 Design Project
Student – Supervisor Agreement

Our signatures below indicate that we have read and understood the following agreement, and that all parties will do their best to live up to the word as well as the spirit of it.

We agree to meet at least once every two weeks for at least half an hour to discuss progress, plans, and problems that have arisen. Before each meeting, the group will prepare a brief progress report that will form the basis for the discussions at the meeting.

If a meeting has to be cancelled by the supervisor, she/he should advise the group as early as possible. If a student cannot attend a meeting, she/he should advise members of the group as well as the supervisor as early as possible.

Both the supervisor and the students will:

- Inform themselves of the course expectations and grading procedure.

The supervisor will:

- Provide regular guidance, mentoring, and support for his/her design project group(s),
- Take an active role in evaluating the work and performance of the students' by completing the supervisor's portion of the grading forms for each course deliverable expediently.
- Return a photocopy of the completed grading evaluation forms to the appropriate section administrator in a timely fashion.
- Be aware of the aims and processes of the course as outlined in the Supervisor's Almanac.

We have read and understood this agreement. Date: Oct 26, 2016

Signature of supervisor: _____

Signature of student: _____

Signature of student: [Signature]

Signature of student: [Signature]

Signature of student: [Signature]

Last revision: 7/08

6.2 Appendix B: Project Proposal Document Attribution Table

This is the Project Proposal Document Attribution Table.

Abbreviation Codes are listed below.

Section	Student Initials		
	<i>CS</i>	<i>EV</i>	<i>YZ</i>
Executive Summary		<i>RD</i>	
Background and Motivation	<i>RD</i> <i>MR</i>	<i>RS</i> <i>MR</i>	<i>RS</i> <i>MR</i>
Goal		<i>RD</i>	
Requirements	<i>RD</i>	<i>RD</i>	<i>RD</i>
Tests	<i>MR</i>	<i>RS</i>	<i>RD</i>
Solutions & Design Alternatives		<i>RS</i> <i>RD</i> <i>MR</i>	<i>RS</i> <i>RD</i>
System-level Overview	<i>MR</i>		<i>RD</i>
Module-level Descriptions	<i>OR1</i>		<i>RD</i>
Assessment of Proposed Solution			<i>RD</i> <i>MR</i>
Work Breakdown Structure & Gantt Chart	<i>RS</i> <i>MR</i>		<i>RD</i>
Financial Plan		<i>RD</i>	<i>ET</i>

Feasibility Assessment	<i>MR</i> <i>RS</i> <i>RD</i>	<i>RS</i> <i>RD</i>	
Conclusion		<i>RD</i>	
Appendix		<i>RD</i>	
<i>All</i>	<i>ET</i> <i>FP</i> <i>CM</i>	<i>ET</i> <i>FP</i>	<i>ET</i> <i>FP</i>

Abbreviation Codes:

Fill in abbreviations for roles for each of the required content elements. You do not have to fill in every cell. The “**All**” row refers to the complete document and should indicate who was responsible for the final compilation and final read through of the completed document.

RS – responsible for research of information

RD – wrote the first draft

MR – responsible for major revision

ET – edited for grammar, spelling, and expression

OR – other

“All” row abbreviations:

FP – final read through of complete document for flow and consistency

CM – responsible for compiling the elements into the complete document

OR - other

If you put OR (other) in a cell please put it in as OR1, OR2, etc. Explain briefly below the role referred to:

OR1: Wrote the module for database.

Signatures

By signing below, you verify that you have read the attribution table and agree that it accurately reflects your contribution to this document.

<i>Na me</i>		<i>Signa ture</i>		<i>Da te:</i>	
<i>Na me</i>		<i>Signa ture</i>		<i>Da te:</i>	
<i>Na me</i>		<i>Signa ture</i>		<i>Da te:</i>	
<i>Na me</i>		<i>Signa ture</i>		<i>Da te:</i>	

6.3 Appendix C: Glossary

This is a glossary of all the technical terms that we used in the context of MovieQA.

Natural language - Colloquial language used by humans in everyday interactions.

Semantics - High-level meaning; here this refers to understanding human actions in the contexts of motivation, intent, and emotion [9]

Semantic Similarity (NLP) - this refers to how related the meaning of words are. Ideally a vector representation of words would place words with higher semantic similarity close together, and low semantic similarity further apart.

Recurrent Neural Network - A class of generally shallow neural networks (single to very few hidden layers) which have memory; current outputs are based on previous states of the system. These are ideal for sequenced inputs/outputs, such as reading and outputting text. The current state of the hidden layer depends on previous states as given by

$$\begin{aligned} h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\ y_t &= W_{hy}h_t \end{aligned} \quad [31],$$

where h_t is the state of the hidden layer at time t , x_t is the input, y_t is the output, and the W 's are parameters learned by the model

Convolutional Neural Nets - A class of neural networks which uses sets of convolutional layers to observe different features; layers are produced by convolving a filter (to select the feature of interest) over the entire image or field under observation. The outputs of these layers are then passed to a nonlinear activation layer, using functions such as RELU or sigmoid. For classification tasks, the final layer is a classifier such as softmax, which produces probability distributions for each class.

Long Short Term Memory (LSTM) - A RNN which uses cell states to update values, mitigating the vanishing gradient effect. The cell states C and hidden state h are given by

$$\begin{aligned} c_t^l &= f \odot c_{t-1}^l + i \odot g \\ h_t^l &= o \odot \tanh(c_t^l) \end{aligned} \quad [32].$$

The flow of information from the cell to the hidden layer is controlled by a series of gates: f acts as a “forget gate”, which can erase the current cell state; i is the “input gate” which determines which states are updated; “ g ” which is the factor added to the cell states; and “ o ” is the “output gate” which determines which cell states are passed to the hidden layer .

Skip-gram - See word2vec

CBOW - See word2vec

Subtitles - Text descriptions of dialogue and sound, often used by the hearing impaired.

DVS - (Descriptive Video Service) Narration of visual information, often used by the visually impaired.

Plot synopsis - A summarization of key events in the film, focusing on events and character action rather than visual description. The MovieQA paper takes from the IMDB website, where anyone can write and edit the summaries [9].

Softmax classifier -

1-hot encoding - all entries in a vector are 0 except for the entry corresponding to the index of the object, which is a 1.

Word2Vec- Word2vec uses a two layer neural network to map words to an n dimensional encoding space. The goal is to map similar words to vectors that are closer together. To do this, the network is trained using word to context-word pairs. When finished, the weights of the model can be interpreted as vector representations of each word.

Skipthoughts -

The encoding/decoding is given by the following equations in Figure 8:

$$\begin{aligned} \mathbf{r}^t &= \sigma(\mathbf{W}_r \mathbf{x}^t + \mathbf{U}_r \mathbf{h}^{t-1}) \\ \mathbf{z}^t &= \sigma(\mathbf{W}_z \mathbf{x}^t + \mathbf{U}_z \mathbf{h}^{t-1}) \\ \bar{\mathbf{h}}^t &= \tanh(\mathbf{W} \mathbf{x}^t + \mathbf{U}(\mathbf{r}^t \odot \mathbf{h}^{t-1})) \\ \mathbf{h}^t &= (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t \end{aligned} \quad \begin{aligned} \mathbf{r}^t &= \sigma(\mathbf{W}_r^d \mathbf{x}^{t-1} + \mathbf{U}_r^d \mathbf{h}^{t-1} + \mathbf{C}_r \mathbf{h}_i) \\ \mathbf{z}^t &= \sigma(\mathbf{W}_z^d \mathbf{x}^{t-1} + \mathbf{U}_z^d \mathbf{h}^{t-1} + \mathbf{C}_z \mathbf{h}_i) \\ \bar{\mathbf{h}}^t &= \tanh(\mathbf{W}^d \mathbf{x}^{t-1} + \mathbf{U}^d(\mathbf{r}^t \odot \mathbf{h}^{t-1}) + \mathbf{C} \mathbf{h}_i) \\ \mathbf{h}_{i+1}^t &= (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t \end{aligned}$$

Figure 8: Encoding Equations (left), Decoding Equations (right)

TF-IDF- TF-IDF is composed of two functions:

- a. Term Frequency: the frequency of a word in a document. A document with a higher frequency is more likely to be related. It is generally calculated as:

$$TF = \frac{\text{frequency of a word in document}}{\text{\#of unique words in document}}$$

- b. Inverse Document Frequency: the number of distinct documents a word appears in. A word with higher inverse frequency carries less weight as it is generally discounted as a stock word (e.g., “and”, “the”, and “of”).

This can be done as follows:

$$IDF = \log\left(\frac{\text{\# of documents}}{\text{\# of documents the word appears in}}\right)$$

Therefore, the score for relevance is:

$$TFIDF \text{ Score} = TF * IDF$$

where a higher score indicates higher relevance.

Stemming - Stemming involves replacing a word with its basic root word; for instance replacing “running” with “run”